Rapport

TP2 Big Data Analytics

auteurs:

Ahmad HARKOUS
Thomas FERMELI FURIC

mars, 2024

Table des matières

1 Exercice 2

1.1 Question 1

Nous obtenons un total de **39 656 098** trajets dans le jeu de données.

1.2 Question 2

Le plus petit montant enregistré est **$-2567,8** et le plus grand montant est **$401095.62**.

1.3 Question 3

Après avoir filtré les données pour ne garder que les montants positifs et inférieurs à **$1000, 255 789** enregistrements ont été retirés.

1.4 Question 4

Après avoir calculé quelques statistiques sur les distances des trajets, nous obtenons que la distance minimale est 0.0 mile et la distance maximale est **389678.46 miles**. La distance moyenne des trajets est **5.96 miles**.

1.5 Question 5

Après avoir filtré les données pour ne garder que les trajets de distance positive et inférieure à 100 miles, **539 413 enregistrements** ont été retirés.

1.6 Question 6

Après avoir filtré les données pour ne garder que les trajets avec un nombre de passagers **entre 1 et 10, 2 041 852 enregistrements** ont été retirés.

1.7 Question 7

Voici la distribution du nombres de passagers :

Nombre de passagers	Nombre de trajets
1.0	27,710,355
2.0	5,767,216
3.0	1,514,075
4.0	690,128
5.0	680,915
6.0	456,203
7.0	80
8.0	55
9.0	17

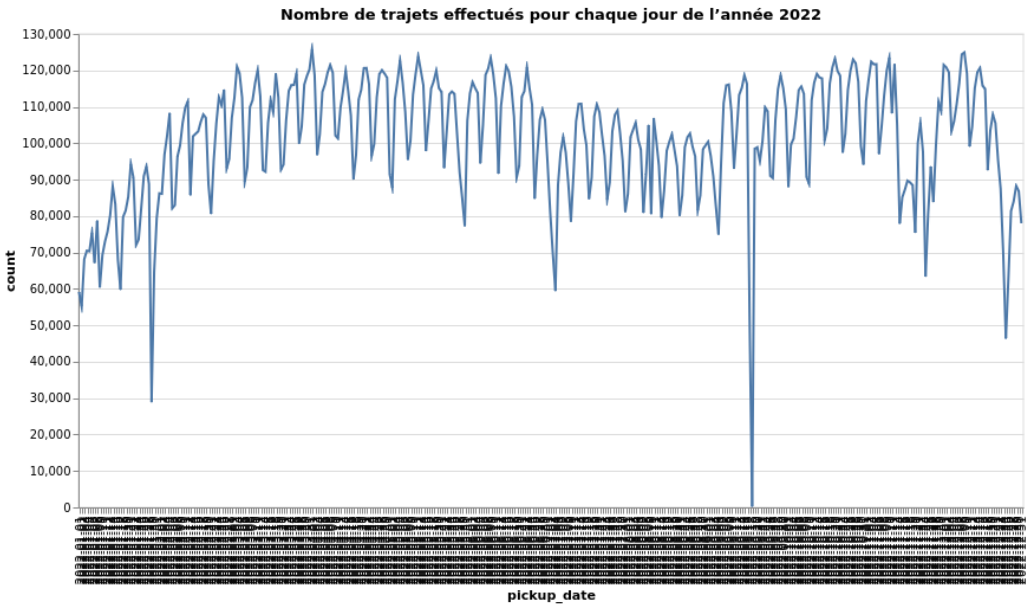
1.8 Question 8

Après avoir filtré les données pour ne garder que les trajets dont la date de début et de fin est en 2022, **1025 enregistrements** ont été retirés.

2 Exercice 3

2.1 Questions 1 et 2

Voici le nombre de trajets effectués pour chaque jour de l'année :

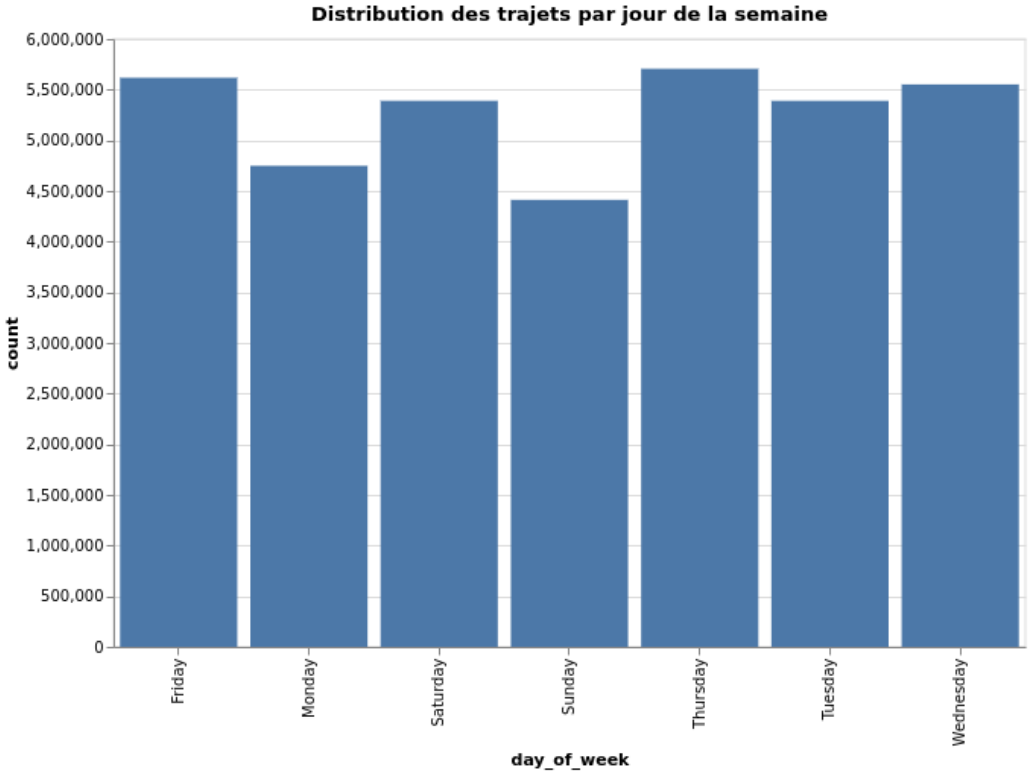


Les dates se superposent sur l'axe des abscisses car nous avons dû contracter le graphe pour qu'il ne soit pas trop large à l'affichage. Nous pouvons voir trois jours pour lesquels il y a eu beaucoup moins de trajets : le 25/12 car c'est

le jour de Noël, le 29/01 pour une raison que nous ignorons car évènement particulier ne se passe ce jour là aux USA, et enfin le 18/09 mais le nombre de trajets est si faible (58 trajets) que nous supposons qu'il y a eu une erreur dans la collection ou la saisie des données.

2.2 Questions 3 et 4

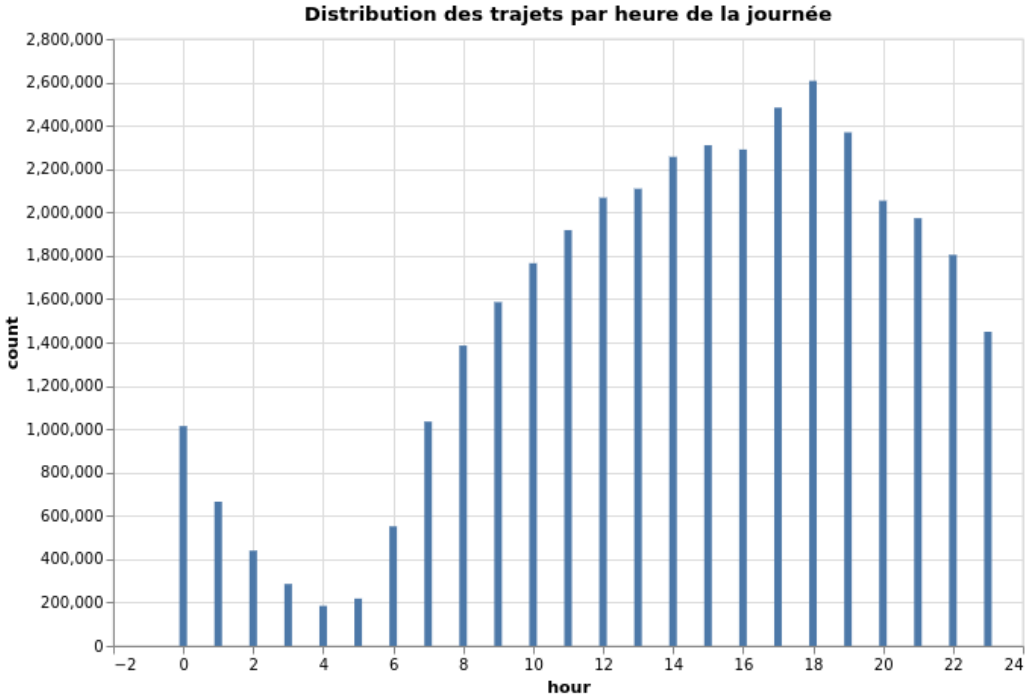
Voici la distribution du nombre de trajets par jour de la semaine :



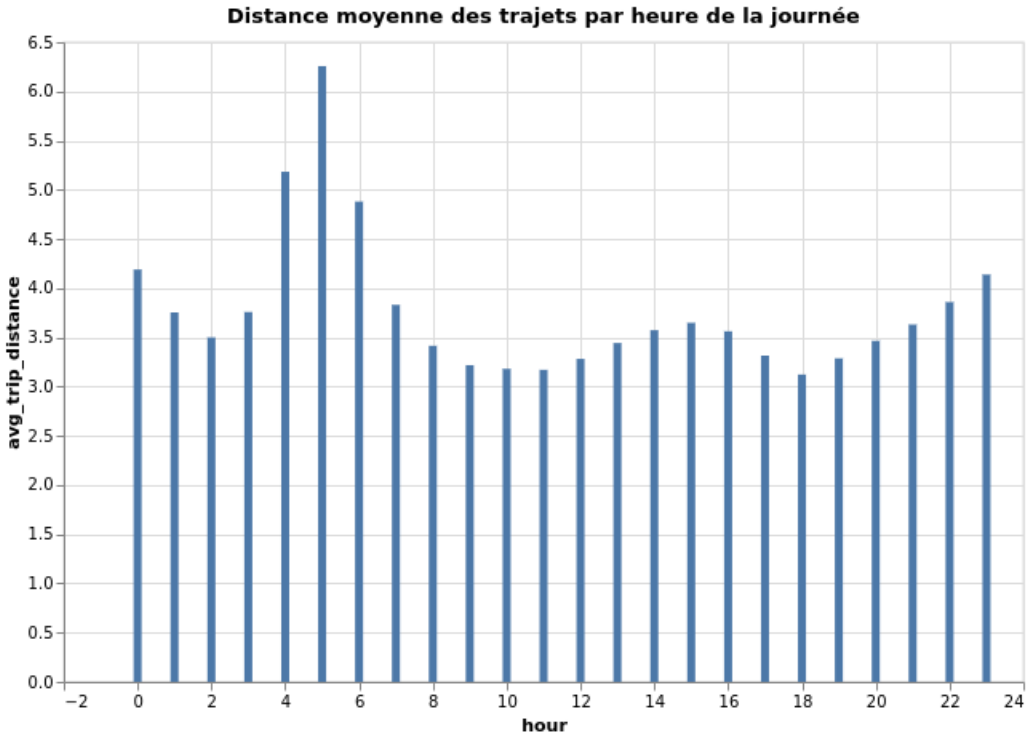
Nous pouvons voir que le nombre de trajets est assez constant tout au long de la semaine, sauf le dimanche et le lundi où les gens sont moins mobiles.

2.3 Questions 5 et 6

Voici le nombre de trajets pour chaque intervalle d'heure :



Voici la distance moyenne des trajets pour chaque intervalle d'heure :

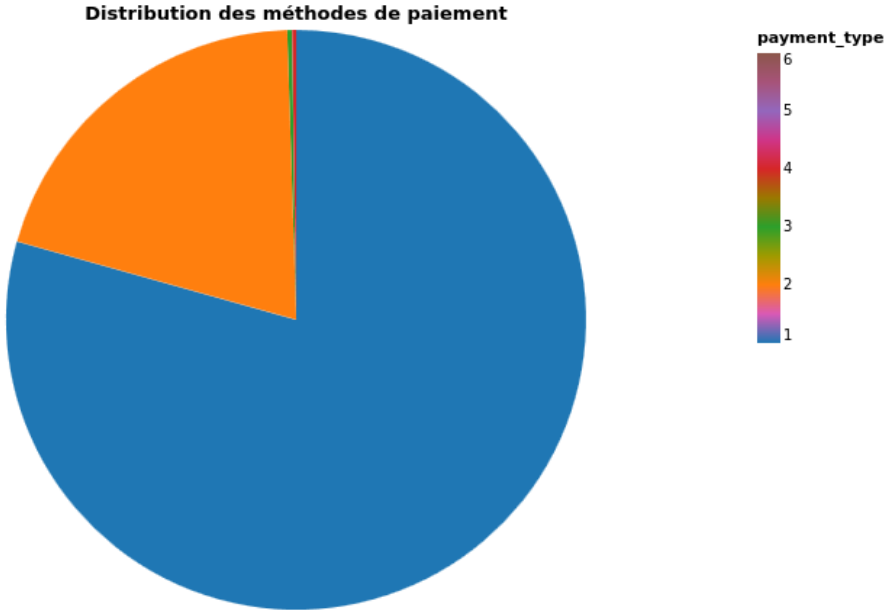


Nous pouvons voir que les trajets les plus courts se produisent tout au long

de la matinée et le soir de 18h à 19h. C'est probablement parce que les gens vont et rentrent du travail sur ces heures-là, contrairement aux autres trajets qui peuvent être plutôt exeptionnelles donc plus longs car les gens sont sortis pour assister à des évènements qui ne se trouvaient pas proches de chez eux.

2.4 Question 7

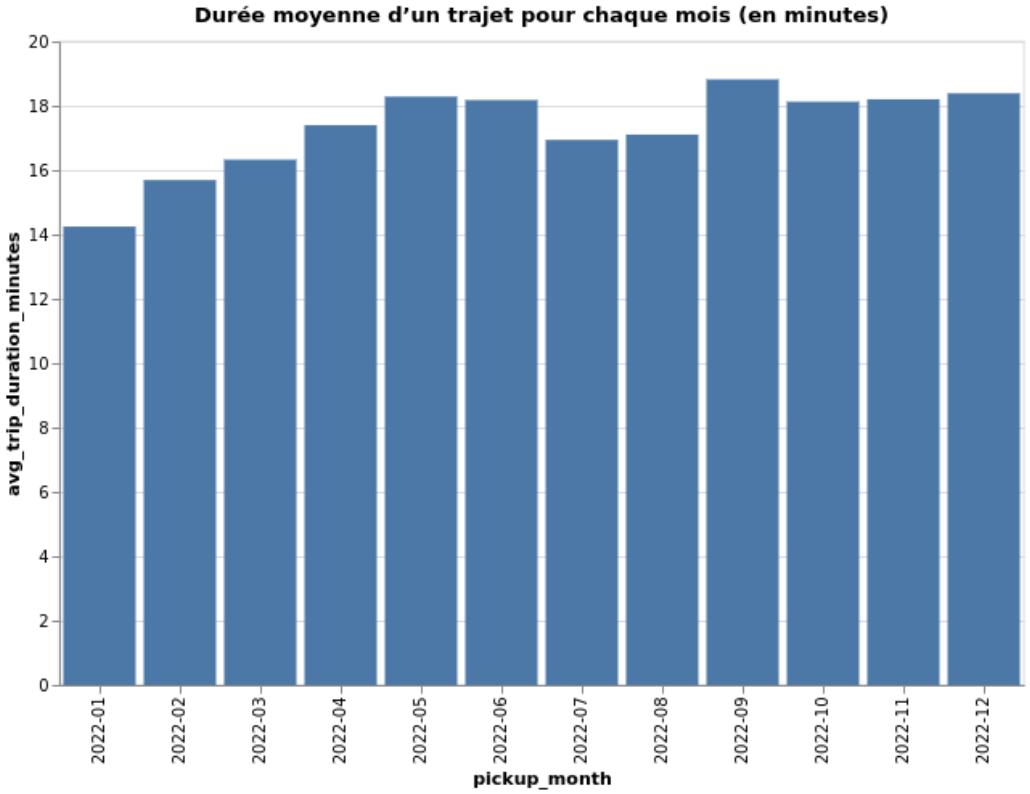
Voici la distribution des moyens de paiement :



Ici nous voyons que 80% des paiements s'effectuent par carte bancaire et 20% par cash.

2.5 Question 8

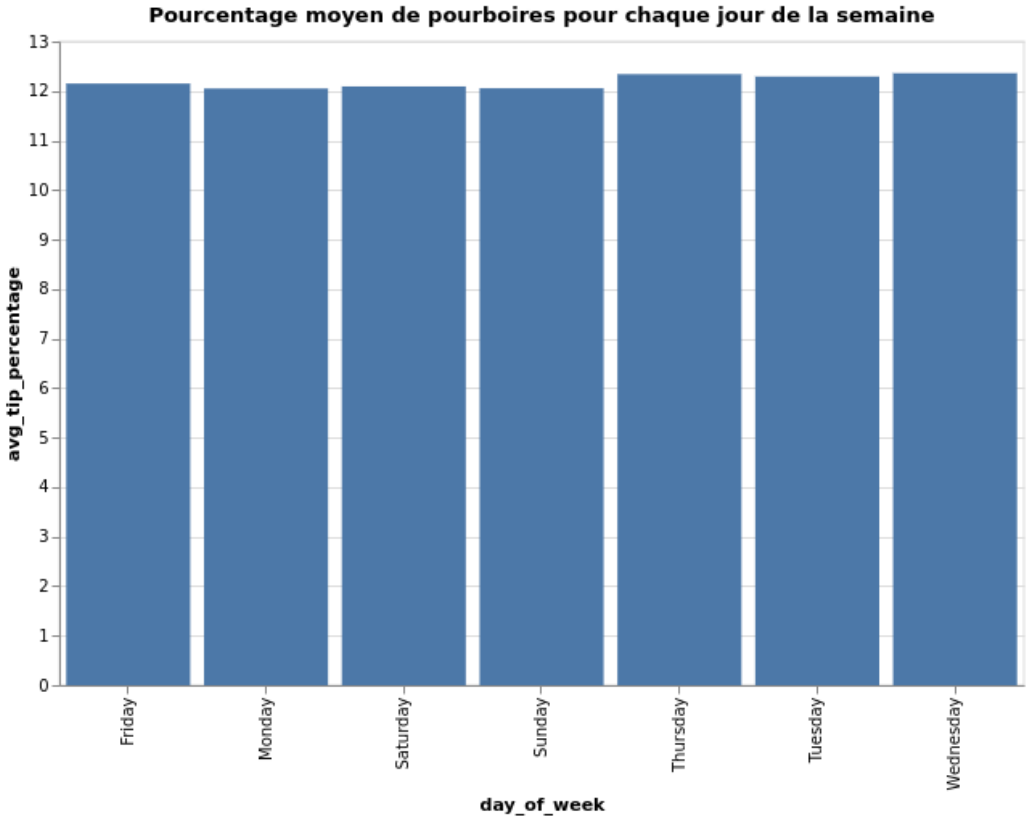
Voici la durée moyenne des trajets pour chaque mois :



Nous voyons que les trajets durent moins longtemps en hiver, probablement car les gens prennent le taxi pour des trajets plus courts pour ne pas affronter le froid.

2.6 Question 9

Voici la moyenne du taux de pourboire donné à chaque trajet par jour de la semaine :



Nous voyons qu'aucune tendance ne se dégage, les clients donnent en moyenne 12% de pourboire à la fin d'un trajet, peu importe le jour de la semaine

3 Exercice 4

3.1 Question 1

La solution c'est de joindre les 2 datasets en un seul dataframe qui contient les colonnes PULocationID—PUBorough— PUZone—PUservice_zone—DOLocationID—DOBorough—DOZone—DOservice_zone

- **Jointure pour l'emplacement de prise en charge (PU) :** Les données de trajet de taxi sont jointes avec les données de localisation des zones de taxi en utilisant la colonne PULocationID, qui représente l'emplacement de prise en charge. Les informations sur l'arrondissement et la zone sont ajoutées au DataFrame résultant.
- **Jointure pour l'emplacement de dépose (DO) : Le DataFrame résultant de l'étape précédente** est à nouveau joint avec les données

de localisation des zones de taxi, cette fois en utilisant la colonne DOLocationID, qui représente l'emplacement de dépose. Les informations sur l'arrondissement et la zone sont ajoutées au DataFrame final.

Vous pouvez trouver le schéma du dataframe dans *ex4/q1/q1.output.txt*.

3.2 Question 2

DOZone	trip_count
Upper East Side N...	123470
Upper East Side S...	106355
Lenox Hill West	78341
Upper West Side S...	75050

Table 1: Top 4 destination zones based on trip count

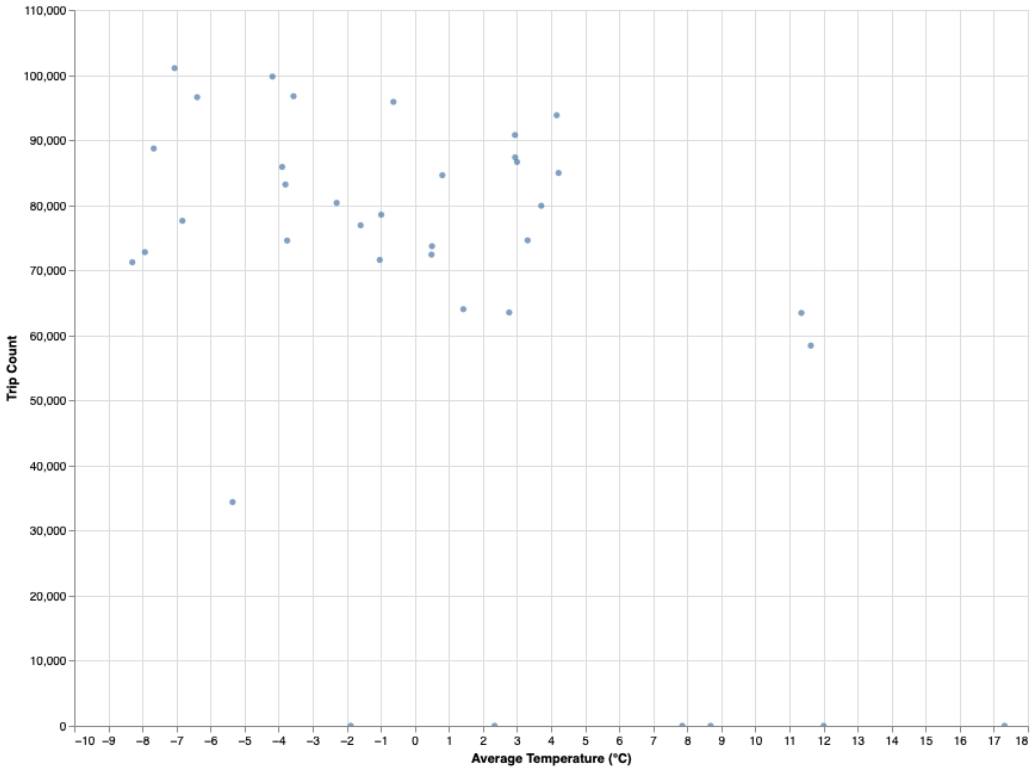
PUZone	trip_count_from_starting_zone
Upper East Side N...	42687
Upper East Side S...	40729
Lenox Hill West	20108
Yorkville West	18563
Lincoln Square East	18371

Table 2: Top 5 Starting zones for top 4 destination zones

4 Exercice 5

Un fichier Jupyter Notebook disponible à l'emplacement *ex5/q1.ipynb* contient toutes les étapes intermédiaires, depuis l'extraction des données météorol ogiques en RDD jusqu'à leur conversion en DataFrame et leur jointure avec le DataFrame des trajets de taxi.

La visualisation interactive du graphique est possible en ouvrant le fichier *ex5/scatter_plot.html* dans un navigateur.
L'axe des abscisses représente la moyenne de la température en degrés Celsius pour un jour en 2022.



Nous observons, à partir de l'analyse du graphique, que les gens ont tendance à prendre plus les taxis en hiver, lorsque les températures sont basses.
Cependant, il y a quelques **anomalies** dans les données. Pour les trajets en taxi, certains jours de l'année 2022 présentent très peu de trajets, par exemple de 1 à 5 trajets, ce qui ne semble pas normal pour une ville comme New York.

En ce qui concerne les données météorologiques, certaines températures sont manquantes, remplacées par le chiffre **9999**. Il ont été filtrer ultérieurement avant le traitement des données.

5 Exercice 6

5.1 Pré-traitement des données

Pour cet exercice, nous avons réutilisé le jeu de données nettoyé que nous avions sauvegardé à la fin de l'exercice 2. Nous avons ensuite créé une colonne **trip_duration** grâce à un calcul sur les colonnes **tpep_pickup_datetime** et **tpep_dropoff_datetime**. Au final, nous n'avons gardé que les colonnes **trip_distance** et **trip_duration** pour les features et la colonne **total_amount**

pour le label. Nous avons également divisé ces données en un jeu d'entraînement (80%) et un jeu de test (20%).

5.2 Choix et entraînement du modèle

La corrélation entre nos données est assez évidente, plus un trajet sera long en distance et en temps, plus le prix sera élevé. Nous avons donc choisi d'utiliser un modèle de **régression linéaire**. Nous avons évalué ses performances grâce à la **Root Mean Squared Error** (RMSE), qui est la métrique classique pour un problème de régression.

5.3 Validation

Lorsque nous évaluons le modèle sur le jeu de test, nous obtenons une RMSE d'environ 6, ce qui signifie qu'en moyenne notre modèle prédit le montant d'une course avec un écart de $6. Ce montant n'est pas très représentatif car il y a une grande diversité dans les ordres de grandeur des montants du jeu de données.

Nous avons donc créé un nouvel enregistrement afin de tester notre modèle. En visualisant le jeu de données, nous pouvons voir un enregistrement dont la distance parcourue est 3.8 miles, le temps de trajet est 18 minutes et le montant total de la course est $21.95. Nous avons donc créé un enregistrement dont la distance est 4.0 miles, la durée est 20 minutes et le modèle a prédit un montant de $23 pour cette course, ce qui paraît complètement cohérent.

6 Exercice 7

6.1 Question 1

Pour charger les données des films et des évaluations, nous utilisons la méthode spark.read.csv avec le paramètre **sep="::"**. La méthode .toDF("col1", "col2", "coln") est ensuite appliqué pour crée un DataFrame à partir des données lues L'output de notre programme, qui contient le dataframe, peut être trouvé dans *ex7/q1/q1.py*.

6.2 Question 2

Le jeu de données contient un total de **10 681** films.

6.3 Question 3

Pour créer une nouvelle colonne contenant l'année de sortie de chaque film à partir du titre, nous avons utilisé la fonction **regexp_extract** sur la colonne "title" avec l'expression régulière:

```
\((\d{4})\)$
```

Cela nous permet d'extraire l'année entre parenthèses à la fin de chaque titre de film. Vous pouvez trouver la sortie de notre programme ainsi que le schéma du dataframe avec la nouvelle colonne dans *ex7/q3/q3.output.txt*.

6.4 Question 4

Voici la liste de tous les genres de films disponibles:

- Crime
- Romance
- Thriller
- Adventure
- Drama
- War
- Documentary
- Fantasy
- Mystery
- Musical
- Animation
- Film-Noir
- IMAX
- Horror
- Western
- Comedy

- Children
- Action
- Sci-Fi

Il existe un film pour lequel le genre n'est pas répertorié **(no genres listed)**

6.5 Question 5

Vous pouvez trouver la liste des films pour chaque genre dans le répertoire *ex7/q5/genres*, avec un fichier txt pour chaque genre.

6.6 Question 6

Rating	Count
0.5	94988
1.0	384180
1.5	118278
2.0	790306
2.5	370178
3.0	2356676
3.5	879764
4.0	2875850
4.5	585022
5.0	1544812

Table 3: Nombre de films pour chaque appréciation

6.7 Question 7

Count	Title
34864	Pulp Fiction (1994)
34457	Forrest Gump (1994)
33668	Silence of the Lambs, The (1991)
32631	Jurassic Park (1993)
31126	Shawshank Redemption, The (1994)
29154	Braveheart (1995)
28951	Fugitive, The (1993)
28948	Terminator 2: Judgment Day (1991)
28566	Star Wars: Episode IV - A New Hope (1977)
27035	Apollo 13 (1995)

Table 4: Les 10 films les plus regardés

A Annexes